

**AMENDMENTS TO CLAIMS**

Kindly amend claims 1, 5, 9 and 13 as follows.

1. (Currently Amended) In a data processing system having a memory, an operating system executing within said data processing system comprising:

    a debug support module configured to associate a debug flag with debug commands issued within the data processing system; and

    a kernel module within said data processing system coupled for communication with said debug support module, said kernel module comprising:

        a process creation unit configured to spawn special processes with a debug flag set for said issued debug commands associated with a debug flag issued wherein a debug flag indicates a process is a debug process with access to debug resources, and

        a messaging transfer unit configured to transfer messages from a source process within said data processing system to a destination process within said data processing system, said message transfer unit further configured to set a debug flag for said destination process responsive to said source process having said debug flag set.

2. (Previously Amended) The operating system of claim 1, wherein said kernel further comprises a memory management unit configured to allocate the memory into a main memory pool and a reserve memory pool, said memory management unit further configured to allocate memory from said reserve memory pool only to said special processes having said debug flag set.

3. (Previously Amended) The operating system of claim 2, wherein said memory management unit is further configured to allocate memory to processes from said main memory pool, said memory management unit further configured to allocated memory to said special processes from said reserve memory pool responsive to said main memory pool is depleted and said debug flag of said special process is set.

4. (Previously Amended) The operating system of claim 1, wherein said process creation unit is further configured to spawn regular processes for commands issued which lack a debug flag, said regular processes lacking a debug flag indicator.

5. (Currently Amended) In a data processing system having a memory, a method for inheriting memory management policies from a source process to a destination process comprising:

receiving a message for transfer from the source process to the destination process within said data processing system;

determining if said source process is associated with a debug flag within said data processing system wherein a debug flag indicates that a process is a debug process with access to debug resources;

associating a debug flag with said destination process responsive to said source process is associated with a debug flag within said data processing system; and communicating the message to the destination process within said data processing system.

6. (Previously Amended) The method of claim 5 further comprising:

determining if a debug command is issued within the data processing system;

spawning a new process associated with said debug command within said data processing system; and

associating a debug flag with said new process to identify said new process as a debug process within said data processing system.

7. (Previously Amended) The method of claim 5, further comprising:
  - allocating the memory into a main memory pool and a reserve memory pool;
  - receiving a memory allocation request from a requesting process within said data processing system; and
  - allocating memory to said requesting process from the main memory pool within said data processing system.
8. (Previously Amended) The method of claim 7, further comprising:
  - determining if said main memory pool is depleted within said data processing system;
  - determining whether said requesting process is associated with a debug flag within said data processing system; and
  - allocating memory to said requesting process from the reserve memory pool if responsive to said main memory pool is being depleted and said requesting process being associated with a debug flag within said data processing system.
9. (Currently Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for inheriting memory management policies from a source process to a destination process in a data processing system having a memory, said method comprising:

receiving a message for transfer from the source process within said data processing system to the destination process within said data processing system; determining if said source process is associated with a debug flag wherein a debug flag indicates that a process is a debug process with access to debug resources;

associating a debug flag into said destination process responsive to said source process being associated with a debug flag; and  
communicating the message to the destination process.

10. (Previously Amended) The program storage device of claim 9, said method further comprising:

determining if debug command is issued within the data processing system;  
spawning a new process associated with the debug command within said data processing system; and  
associating a debug flag with said new process to identify said new process as a debug process.

11. (Previously Amended) The program storage device of claim 9, said method further comprising:

allocating the memory into a main memory pool and a reserve memory pool;  
receiving a memory allocation request from a requesting process within said data processing system;  
allocating memory to said requesting process from the main memory pool within said data processing system.

12. (Previously Amended) The program storage device of claim 11, said method further comprising:

determining if said main memory pool is depleted;  
determining if said requesting process is associated with a debug flag; and  
allocating memory to said requesting process from the reserve memory pool responsive to said main memory pool being depleted and said requesting process being associated with a debug flag.

13. (Currently Amended) In a data processing system having a memory, an operating system executing within said data processing system comprising:

means for receiving a message for transfer from a source process within said data processing system to a destination process within said data processing system;  
means for determining if said source process is associated with a debug flag  
wherein a debug flag indicates that a process is a debug process with access to debug resources;  
means for associating a debug flag into said destination process within said data processing system responsive to said source process being associated with a debug flag; and  
means for communicating the message to the destination process within said data processing system.

14. (Previously Amended) The operating system of claim 13 further comprising:

means for determining if a debug command is issued within the data processing system;

means for spawning a new process within said data processing system associated with the debug command; and

means for associating a debug flag with said new process to identify said new process as a debug process within said data processing system.

15. (Previously Amended) The operating system of claim 13, further comprising:  
means for allocating the memory into a main memory pool and a reserve memory pool;  
means for receiving a memory allocation request from a requesting process within said data processing system;  
means for allocating memory to said requesting process from the main memory pool.
16. (Previously Amended) The operating system of claim 15, further comprising:  
means for determining if said main memory pool is depleted;  
means for determining if said requesting process is associated with a debug flag;  
and  
means for allocating memory to said requesting process from the reserve memory pool if responsive to said main memory pool is being depleted and said requesting process being associated with a debug flag.